In The Claims:

Please amend claims 2, 3, 28, and 29 for clarity, as indicated below.

1. (Previously presented) A system, comprising:

a processor; and

memory coupled to the processor and configured to store program instructions executable by the processor to implement a class structure based data object enhancer configured to:

input one or more classes;

analyze the structure of the one or more classes to determine a persistence structure specifying data fields of the one or more classes to be persisted; and

generate one or more enhanced classes corresponding to the one or more classes such that an object of the one or more enhanced classes is enhanced to persist data of the data fields to be persisted according to the persistence structure, wherein said data of the data fields to be persisted is data of said object.

2. (Currently amended) The system as recited in claim 1, wherein to analyze the structure of the one or more classes, the class structure based enhancer is configured to make one or more Java reflection calls to the one or more classes.

3. (Currently amended) The system as recited in claim 1, wherein to analyze the structure of the one or more classes, the class structure based enhancer is configured to parse bytecode of the one or more classes to determine class and field attributes.

4. (Previously presented) The system as recited in claim 1, wherein the class structure based enhancer is further configured to generate metadata that includes results of the analysis of the structure of the one or more classes.

5. (Original) The system as recited in claim 4, wherein the generated metadata is output explicitly as a metadata file.

6. (Original) The system as recited in claim 5, wherein the metadata file is an extensible markup language (XML) file.

7. (Original) The system as recited in claim 1, wherein the persistence structure corresponds to the structure of the one or more classes.

8. (Previously presented) The system as recited in claim 1, wherein the persistence structure maps the data fields to be persisted to a single table in a database.

9. (Original) The system as recited in claim 1, wherein to determine a persistence structure for the data of the one or more classes the class structure based enhancer is configured to apply one or more rules to the results of Java reflection calls to or byte code parsing of the one or more input classes.

10. (Previously presented) The system as recited in claim 9, wherein the one or more rules applied by the class structure based enhancer include persisting class fields that are not static or transient.

11. (Previously presented) The system as recited in claim 9, wherein the one or more rules applied by the class structure based enhancer include storing persistent fields of a given class in a table corresponding to that class in a database.

12. (Original) The system as recited in claim 1, wherein the one or more classes are comprised in a Java ARchive (JAR) file.

13. (Previously presented) The system as recited in claim 1, wherein the class structure based enhancer is further configured to output the enhanced one or more classes and a database schema for storing the data to be persisted in a persistent data store.

14. (Previously presented) A computer-implemented method, comprising:

receiving one or more classes;

analyzing the structure of the one or more classes to determine a persistence structure specifying data fields of the one or more classes to be persisted; and

generating one or more enhanced classes corresponding to the one or more classes such that an object of the one or more enhanced classes is enhanced to persist data of the data fields to be persisted according to the persistence structure, wherein said data of the data fields to be persisted is data of said object.

15. (Previously presented) The method as recited in claim 14, wherein said analyzing comprises making one or more Java reflection calls to the one or more classes.

16. (Previously presented) The method as recited in claim 14, wherein said analyzing comprises parsing bytecode of the one or more classes to determine class and field attributes.

17. (Previously presented) The method as recited in claim 14, further comprising generating metadata that includes results of said analyzing.

18. (Previously presented) The method as recited in claim 17, further comprising outputting the metadata explicitly as a metadata file.

19. (Original) The method as recited in claim 18, wherein the metadata file is an XML file.

20. (Original) The method as recited in claim 14, wherein the persistence structure corresponds to the structure of the one or more classes.

21. (Previously presented) The method as recited in claim 14, wherein the persistence structure maps the data fields to be persisted to a single table in a database.

22. (Previously presented) The method as recited in claim 14, wherein said determining of the persistence structure for the data of the one or more classes comprises applying one or more rules to the results of Java reflection calls to or byte code parsing of the one or more input classes.

23. (Previously presented) The method as recited in claim 22, wherein the rules include persisting class fields that are not static or transient.

24. (Previously presented) The method as recited in claim 22, wherein the rules include storing persistent fields of a given class in a table corresponding to that class in a database.

25. (Original) The method as recited in claim 14, wherein the one or more classes are comprised in a Java ARchive (JAR) file.

26. (Previously presented) The method as recited in claim 14, further comprising outputting the enhanced one or more classes and a database schema for storing the Previously presented data to be persisted in a persistent data store.

27. (Previously presented) A computer-accessible storage medium, storing program instructions, wherein the program instructions are computer-executable to:

input one or more classes;

analyze the structure of the one or more classes to determine a persistence structure specifying data fields of the one or more classes to be persisted; and

generate one or more enhanced classes corresponding to the one or more classes such that an object of the one or more classes is enhanced to persist data of the data fields to be persisted according to the persistence structure, wherein said data of the data fields to be persisted is data of said object.

28. (Currently amended) The computer-accessible storage medium as recited in claim 27, wherein to analyze the structure of the one or more classes, the program instructions are executable to make one or more Java reflection calls to the input classes.

29. (Currently amended) The computer-accessible storage medium as recited in claim 27, wherein to analyze the structure of the one or more classes, the program instructions are executable to parse bytecode of the one or more classes to determine class and field attributes.

30. (Previously presented) The computer-accessible storage medium as recited in claim 27, wherein the program instructions are executable to generate metadata that includes results of the analysis of the structure of the one or more classes.

31. (Previously presented) The computer-accessible storage medium as recited in claim 30, wherein the program instructions are executable to output the generated metadata explicitly as a metadata file.

32. (Previously presented) The computer-accessible storage medium as recited in claim 31, wherein the metadata file is an extensible markup language (XML) file.

33. (Previously presented) The computer-accessible storage medium as recited in claim 27, wherein the persistence structure corresponds to the structure of the one or more classes.

34. (Previously presented) The computer-accessible storage medium as recited in claim 27, wherein the persistence structure maps the data fields to be persisted to a single table in a database.

35. (Previously presented) The computer-accessible storage medium as recited in claim 27, wherein to determine a persistence structure for the data of the one or more classes the program instructions are executable to apply one or more rules to the results of Java reflection calls to or byte code parsing of the one or more input classes.

36. (Previously presented) The computer-accessible storage medium as recited in claim 35, wherein the rules include persisting class fields that are not static or transient.

37. (Previously presented) The computer-accessible storage medium as recited in claim 35, wherein the rules include storing persistent fields of a given class in a table corresponding to that class in a database.

38. (Previously presented) The computer-accessible storage medium as recited in claim 27, wherein the one or more classes are comprised in a Java ARchive (JAR) file.

39. (Previously presented) The computer-accessible storage medium as recited in claim 27, wherein the program instructions are executable to output the enhanced one or more classes and a database schema for storing the data to be persisted in a persistent data store.